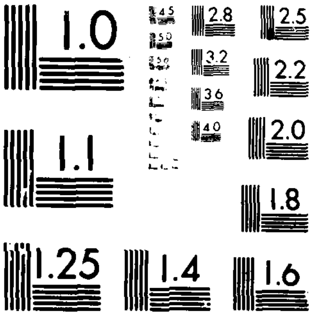
AD-A083 354 FOREIGN TECHNOLOGY DIV WRIGHT-PATTERSON AFB OH F/G 9/2
150(DJS-11) SOFTWARE.(U)
APR 79
UNCLASSIFIED FTD-ID(RS)T-1664-78 NL

1 OF 1
AD
A083354

END
DATE
FILMED
5-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PHOTOGRAPH THIS SHEET

ADA083354

DTIC ACCESSION NUMBER

LEVEL

1

INVENTORY

FTD-ID(RS)T-1664-78

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR

NTIS GRA&I ☑

DTIC TAB □

UNANNOUNCED □

JUSTIFICATION

BY

DISTRIBUTION /

AVAILABILITY CODES

DIST A

AVAIL AND/OR SPECIAL

DISTRIBUTION STAMP

DATE ACCESSIONED

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

FTD-ID(RS)T-1664-78

FOREIGN TECHNOLOGY DIVISION

AIR FORCE SYSTEMS COMMAND

150(DJS-11) SOFTWARE

FOREIGN TECHNOLOGY DIVISION

EDITED TRANSLATION

FTD-ID(RS)T-1664-78 17 April 1979

MICROFICHE NR: FTD-79-C-000552

CSL76012743

150(DJS-11) SOFTWARE

English pages: 27

Source: Pei-Ching Ta-Hsueh Ksueh-Pao--Tzu-Jan
Ko Hsueh-Pan, August 1974, pp. 144-157

Country of origin: China
Translated by: LINGUISTIC SYSTEMS, INC.
F33657-78-D-0618
H. P. Lee
Requester: PHE
Approved for public release; distribution unlimited.

THIS TRANSLATION IS A RENDITION OF THE ORIGINAL FOREIGN TEXT WITHOUT ANY ANALYTICAL OR EDITORIAL COMMENT. STATEMENTS OR THEORIES ADVOCATED OR IMPLIED ARE THOSE OF THE SOURCE AND DO NOT NECESSARILY REFLECT THE POSITION OR OPINION OF THE FOREIGN TECHNOLOGY DIVISION.

PREPARED BY:

TRANSLATION DIVISION
FOREIGN TECHNOLOGY DIVISION
WP-AFB, OHIO.

150(DJS-11) Software

DJS-11 Software Division
Electronic Apparatus Plant

The machine of DJS-11 type is a one million cycle large integrated circuit electronic digital computer. It has such good qualities as small size, large storage capacity, high computation speed and its performance is steady, reliable, and power saving. Its basic components are semiconductor integrated circuit and thick-film circuit and its word length has 48 bits. The capacity of the internal storage is 130,000 words and the external storage contains magnetic disk and magnetic tape. Its input-output devices include paper tape photoelectric inputer, line printer, digit-type curve unit, high speed paper tape perforator, console typewriter and special input-output devices totalling 27 pieces of 11 different kinds. DJS-11 also has interruption system and exchanger. The interruption system is used to supervise the external units, to control multitrack programming, to operate console typewriter, to handle trouble if it develops , and to protect the power source. The exchanger is used exclusively for message transmission.

The DJS-11 has three different types of numeration: whole word length floating point, half word length floating point, and integer. An instruction is made of a half word length and one address. There is also one group of change address registers and an in-last and out-first storage area to store return addresses of rotator and initial address of cyclic segments.

In addition to the function of interruption system, cross storage and overlapping working in the hardware equipment, DJS-11 is also equipped with corresponding software. It has a mini-operation system which is designed to help increase machine efficiency and function - it is a DJS-11 multitrack supervision program. And for the users' convenience in using the computer to solve problems, a block structure language compilation and translation program and symbol compilation program have been designed. For machine maintenance, there is a machine inspection program.

In order to achieve a high speed of calculation of one million times per second, DJS-11 must try to solve the contradiction between the high speed computation ability of the calculator and the low access speed of the magnetic core storage, as well as the contradition between the high speed of master machine and the low speed of external units. Facing the problems mentioned above, DJS-11 adopts the following advanced technology and measures.

1) Storage of Multicell Parallel Operation

For increasing computation speed, in addition to the requirement that the computer must increase its own speed, the storage is required to be able to provide operation numbers and instructions on time. The calculator of the DJS-11 employs high speed semiconductor integrated circuits and high speed computation method (such as multi-digit multiplication and shortcut division), so it can reach a speed of calculating of one million per second. But, the storage is made of magnetic core and its access speed is relatively low, so it cannot provide the operation numbers and instructions that are required for calculating one million times per second. This is the main problem which affects the speed of the machine. For overcoming this difficulty, the storage of DJS-11 is divided into four smaller units and the capacity of each unit can store 32,000 words. The unit is called "cell." Since these four cells can work simultaneously, the access speed increases four times more than before. In order to keep each magnetic core cell in maximum busy state, the cells are controlled infividually. By this form of multi-cell working in parallel, the speed is overwhelmingly increased.

2) Advance Controller

After adopting the system of multi-cell parallel operation, if the calculator, just like an ordinary small computer, can only directly deal with storage, and cannot be free from the limitation of the access speed of single magnetic core cell, the efficiency of multi-cell parallel operation, therefore, cannot be brought into full play. For this reason, DJS-11 is equipped with an "advance controller"inside the controller. "Advance" here means that the storage earlier begins to transmit operation numbers and instructions to the register of the advance controller and the controller earlier prepares a set of computation instructions for the calculator, and stores it in the register of the advance controller. Then the calculator can get its operation numbers and computation instruction directly from the advance controller and directly store the computation results back in it. And the advance controller then puts the results into storage. Thus, the calculator is free from the limitation of the access speed of the magnetic core storage and can continuously have operation numbers and computation instructions from the register of the advance controller of which the access speed is high, and consequently its ability of computation with high speed can be fully expressed. The register of the advance controller is something like a fast and small "storage" of the calculator and it functions as a buffer between the storage and the calculator. Since there is an advance controller, the controller and the calculator no more interfere with each other as often happens in some other ordinary machines (controller gets instructions, analyzes instructions, gets numbers, issues control signals - calculator calculates - controller gets instructions, analyzes instructions, gets numbers, issues control signals - ...), and when controller analyzes instructions, the calculator can simultaneously calculate parallel with the controller. One need not wait for the other. The working efficiency of the machine is thereby promoted.

3) Using an Exchanger to Control External Devices

The external device of an ordinary small comput4r is controlled by a controller. When the controller controls external devices, it is not able to control the calculation of the calculator. External devices are mostly made up of low speed electro-mechanical units so they usually delay. In order to keep the controller busy, DJS-11 gives a part of the work of controlling external devices to an exchanger and let the exchanger control external devices and exchange messages with storage. Only at the beginning and end of message exchange does the controller execute simple control over the exchanger. Thus, when external devices are working the controller and the calculator can work simultaneously and a condition of simultaneous functioning of external devices, calculator and controller is thereby created. Under the control of an exchanger, different external devices can work in parallel.

Because of the four-cell storage, advance controller and the exchanger, the logical block diagram of DJS-11 compared with ordinary machines, shows great variability. Among them, the magnetic disk is an external storage, a part of the external device, and the four cells mentioned above are usually called internal storage.

4) Multitrack Operation

The installation of an exchanger in DJS-11 eliminates the burden of controller from controlling external devices, and promotes efficiency. However, when only one problem is being processed, there always occur the following phenomena: when execution reaches a certain degree it is necessary to store part or all of the data from the storage into external storage, and new data from external storage will be taken in to continue processing. Because the speed of external storage is slow, the calculation and control have to stop to allow data exchange between internal and external storage until new data come in. In order to improve this situation, DJS-11 employes multitrack operation method and feeds four arithmatic programs into the internal storage. When one program needs new data of instruction from external devices to continue operation, the machine records "process standing" and goes on to the next problem; when the second problem cannot continue due to some reasons, it also records "process standing" and goes to the third problem and so forth. The machine comes back to the first problem when its use of external devices is finished. This way, the calculation and control of the machine are kept in busy state and high speed processing ability can be realized.

To carry out multitrack operation, the machine not only needs to increase its logical circuit but also has to have a programming system for supervision and assignment, i.e., multitrack programming.

In summary, DJS-11 uses multi-storage in parallel operation and advance control to overcome the major obstacle caused by the slow access speed of the storage during high speed processing. In order to solve the contradiction between the fast main body and slow external devices, an exchanger is installed to supervise the external devices, and some other measures, such as multitrack operation, parallel operation of controller, calculator and external devices and full use of the waiting time of various parts of the machine, altogether make DJS-11 able to have a high speed of calculating one million times per second.

MULTITRACK PROGRAMMING SYSTEM

In order to increase problem solving ability and speed, DJS-11 is equipped with a small operation system i.e. multitrack operation supervision program. Its basic functions are:

1) to control, adjust and supervise multiple-user programs and to carry out multitrack operation.

2) to control operation of language translation and compilation program and symbol compilation program.

3) to supervise and adjust external devices, to control data transmission process of external devices, to modify the input and output message and to make the master machine and external devices working in parallel.

4) to take care of problems when they develop.

5) to receive, analyze, and execute commands from the operator, to display the conditions of operation within the machine, and to continue operation.

In summary we believe that the difference of DJS-11 multitrack programming system from single-track programming system lies in distribution of storage, supervision of external devices and assignment of programs. The multitrack program system of DJS-11 will be discussed in the following sections.

1. Distribution of Storage

In order to carry out multitrack programming system, the main storage of DJS-11 has to store both supervision programs and multi-user programs. The distribution and usage of storage can no longer be arranged by users and they must be managed by the supervision program.

One way of distributing storage is the method of boundary distribution, which means a method of continuous distribution of storage areas.

"Different contradictions can only be solved by different means." According to the different nature of supervision program and user program we divide the main storage areas into supervision program area and user program area. The former occupies the utmost front area and constitutes one thirteenth of the main storage. The following is user program area.

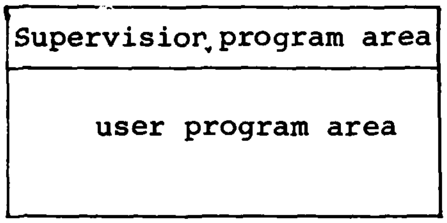


Fig. 1-1

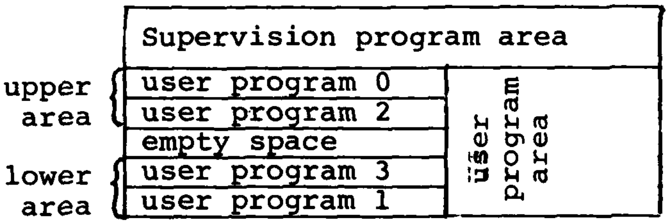


Fig. 1-2

The user program area allows no more than four user programs. How are these four user programs to be arranged so as to provide maximum storage space for the new program and also to minimize the times of moving in the user program area after the withdrawal of the old program? This is one of the problems to be considered in multitrack programming systems. We adopt a principle of placing programs at two ends and leaving the middle empty. The user program area is divided into upper and lower areas. When storing, two areas are stored alternately as shown in Fig. 1-2. When program 2 (or 3) moves out, no remaining program has to move because the vacancy is in the center. When program 0 (or 1) moves out, only one has to move in program 2 (or 3). The advantage of so doing is to avoid a rearrangement of the whole program after one program has withdrawn. But this method has its limit and there is still "moving" especially when there are more than four user programs. The storage rearrangement will increase correspondingly when the number of user programs is increased.

Since more user programs are allowed to store in the main storage at the same time, the storage occupied by each user program has to be dynamically distributed when it is fed into the machine. Therefore, user program is required to be floating. In order to give each user program proper execution and not to be disturbed by any single program which has some mistake, each program in the storage area must be protected.

There are many ways to make floating programs and we use relative addressing methods. The supervision program draws up upper and lower boundaries for the processing program and the boundary addressing storage is set up to accomplish address variation. By the change of lower boundary the programs are made floating. Using boundary crossing protection method to prevent mistakes of the execution program from destructing other programs in the machine.

The advantages of boundary distribution are that the execution is fast and saves equipment. The disadvantages are: the time of occupying the main storage is long, and the change of programs in the machine requires rearrangement of storage area.

Supervision program controls machine operation and it holds the key of maintaining normal function of the machine. So its safety is very important. We use "switch protection" method in main storage supervision program because this method requires less equipment.

2. Supervision of External Devices

The external devices of DJS-11 consist of general and special units totalling 27 pieces of 11 different kinds. They, through the exchanger, exchange messages directly with the main storage and, thorugh interruption system, report to the master machine. Therefore, parallel work is attained between the master machine and external devices and among these devices.

DJS-11 external devices have to serve many user programs, and like the main storage they require "dynamic distribution" of supervision programs. At the same time, the control of external device transmission procedure is directed by the supervision program through interruption systems under "supervised state."

A user program has to use external devices through supervision program, so the machine is designed to have one set of machine instruction as well as one set of general instruction. The instruction is interpreted and executed by the supervision program and is the procedure for the user programs to employ external devices through the supervision program.

In order to supervise external devices, the supervision program establishes a channel schedule for each channel of the exchanger to record (1) conditions of its external device, good or bad, whether distributed, and to whom, (2) channel working condition, busy or idle, which channel (channel no.) and which external device (unit no.) is working.

1) Distribution of external devices

The distribution of external devices is in the form of user requests and supervision program assignment. There are two ways of requesting external devices by the users: group request at the time when user programs are fed into the machine and request during the process of execution. The former needs to be submitted at the beginning of a program, the latter can use general instruction or operation command. A typical request information can be like this:

Need X type equipment as my nth device;
The "nth" device is a relative number of the user's own external devices and it does not necessarily correspond to the number of any one external device before the distribution by supervision programs.

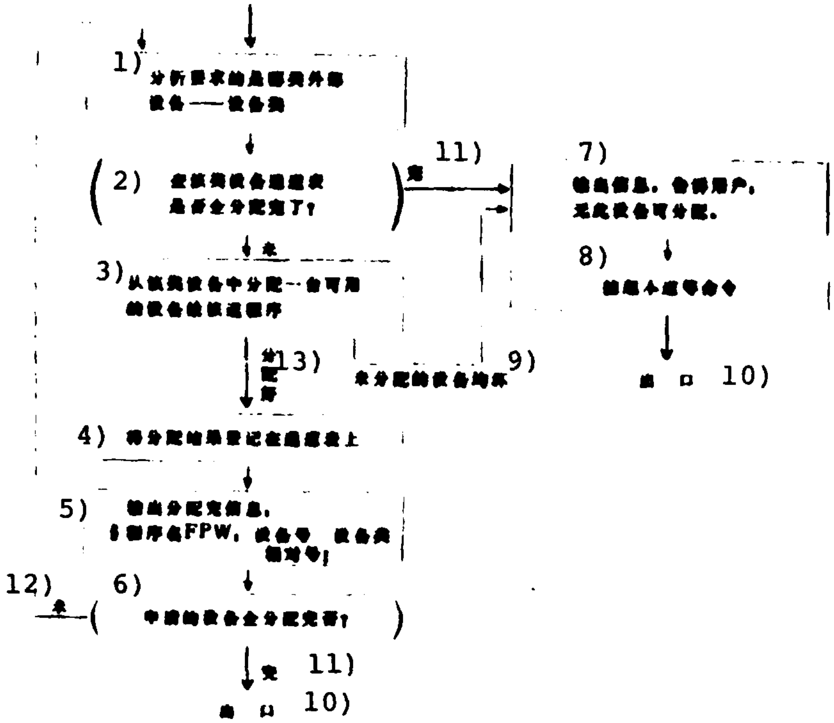
With the help of channel schedule, the distribution by supervision programs can be carried out according to current situations and records the result in the channel schedule:

x equipment of x type is assigned to x channel user program
device no.   type   channel no.
as nth device. This message is also sent out to users through
relative no.
console typewriter.

Program name FPW: device number device type relative number! In which, "FPW" stands for "complete assignment."

In this way, corresponding relationships between relative number of external devices of the user program and its given external device number is established. When writing programs, the user needs only its own relative number as if all external devices can be freely used by it.

The working diagram of external device assignment by supervision program is like this:



1) to analyze what type of external device has been requested - equipment type
2) to check if the channel schedule of that type of equipment is available
3) to assign one unit of that type of equipment to this channel program
4) to record this assignment on the channel schedule,
5) to send out complete assignment message: program name, equipment type, number, relative number.
6) Is all requested equipment assigned?
7) to send message to the user: no such equipment available
8) to suspend this channel, and wait for instruction
9) unusable, not assigned, 10) outlet, 11) yes, 12).no
13) assigned

An external device will belong to the user program once it has been assigned. It cannot be used by any other channel programs before this user finishes using it, so that independence of using external devices by each user program is guaranteed.

In order to yield high efficiency of using external devices, when the user program in external devices has been completed the supervision program should be notified immediately "to assign", and the external devices should also be returned to the supervision program for the use of other programs. There are two ways of returning: one is that the user voluntarily returns with general instruction or operation command; the other is that when the user program withdraws, the supervision program automatically retrieves all the occupied external equipment.

2) Procedures for Users to Use External Devices - General Instruction

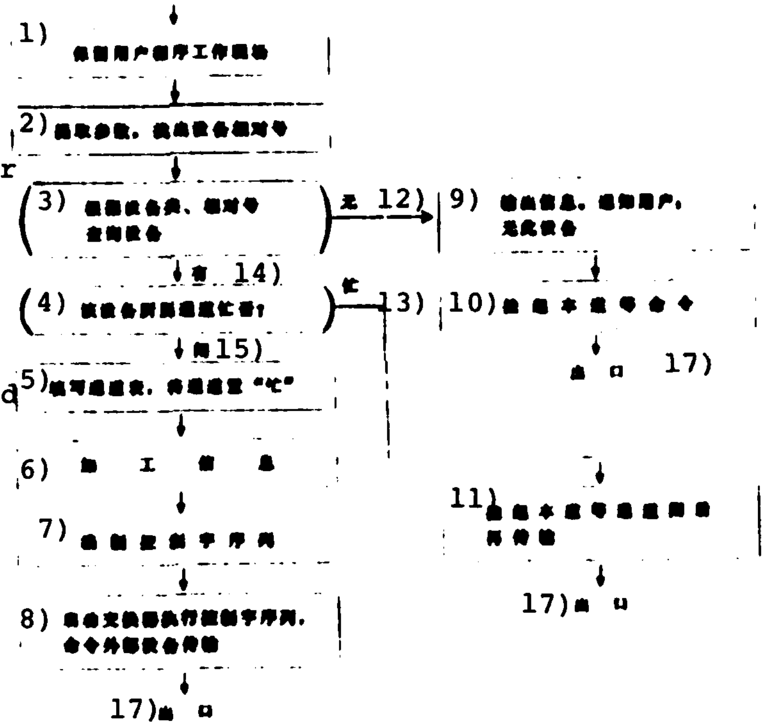
The general instruction of DJS-11 is made up of "transfer" instruction and several parameters. Its general form is:

$$\left\{\begin{array}{l}\uparrow\rightarrow *N\\ \text{Parameter}\end{array}\right.$$

The first line is "transfer" instruction (machine instruction) which is used to transfer from user program to supervision program. The second line is the parameter provided by instruction. There are two parameters for general instruction: type of equipment and relative number. A processing program is set up in the supervision program to correspond to one type of external equipment and one set of general instruction: it is called XX equipment supervision program, such as wide line supervision program. Each general instruction has its corresponding processing program which is a standard process and has "re-entry" function. All processing programs of input-output devices constitute an input-output control system.

3) Transmitting Procedure of External Devices

The control of external device transmitting procedure is carried out by supervision program through interruption system under "supervised state." At the time when user programs are transfered with general instruction into supervision program,

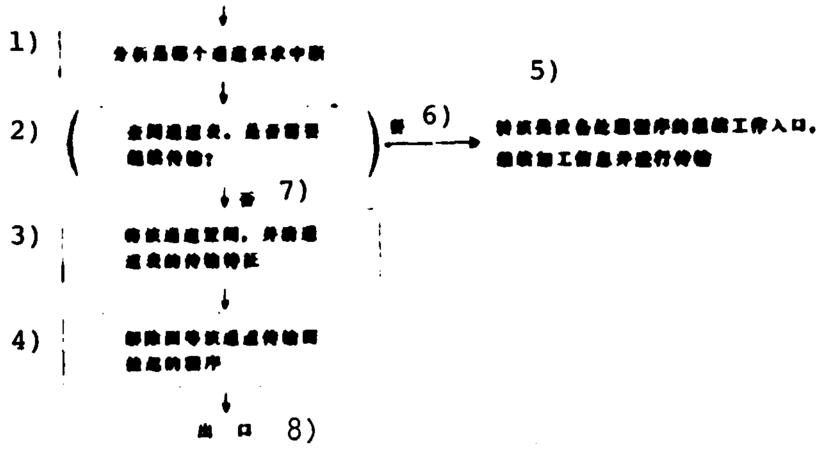


1) maintain user program working status
2) pick up parameter and find out equipment relative number
3) check equipment based on type and relative number
4) is its channel busy?
5) fill channel schedule, get
6) working information
7) arrange control word order

8) start the exchanger to execute control order instruct external device to transmit, 9) send message to users: no such equipment available, 10) suspend and wait for further instruction, 11) suspend until channel is open to transmit, 12) no, 13) busy, 14) available, 15) idle, 17) outlet

the machine is in "supervised state." The processing steps of the supervision program are shown in the above diagram.

When external devices complete their work, they will give a signal of "transmission completed." Then the interrupt processing program will continue to process. The working process is shown in the following diagram.



1) determine which channel requests interruption
2) check channel schedule to see if continuous transmission is needed
3) idle the channel, and give the transmission characteristics of the channel schedule
4) release the suspended program which has been idled due to waiting for transmission
5) Inlet of continuous work of the processing program of that equipment. Continuous working information and transmission
6) Yes, 7) No, 8) Outlet

Both outlets of the above mentioned programs are directed toward a certain part of the interrupt processing program to continue handling other interrupt signals and finally enters channel selection program to select one executable user program.

This ends supervised status and begins print-out.

3. User Program Assignment

1) Assignment Schedule

The assignment schedule is a file established by supervision program for the purpose of assigning each user program, and it reflects the status of user program in the machine. At the time when user program is installed in the machine, supervision program establishes assignment schedule. When user program changes in operation, the schedule is modified. When user program withdraws, the schedule is withdrawn too. A schedule contains program name, channel number, program status, program type, priority number, upper or lower storage boundary, break point, console typewriter number, and work reserve area. Only program status and priority number will be explained in detail in the following.

Program status - The user programs in the machine are different in status. They are used to indicate whether programs are ready to jion the execution. In general, there are three kinds of status: (a) execution status is the moment that the program is being executed; there is only one channel in execution status at any time; (b) waiting status is a program which has execution conditions but is not yet picked for execution by supervision program; and (c) idle status: when user programs for some reasons do not meet execution requirement, they are put in status of suspension. There are waiting for transmission suspension, waiting for command suspension and waiting for moving suspension. When the cause of suspension is diminished, they will change to be in waiting status.

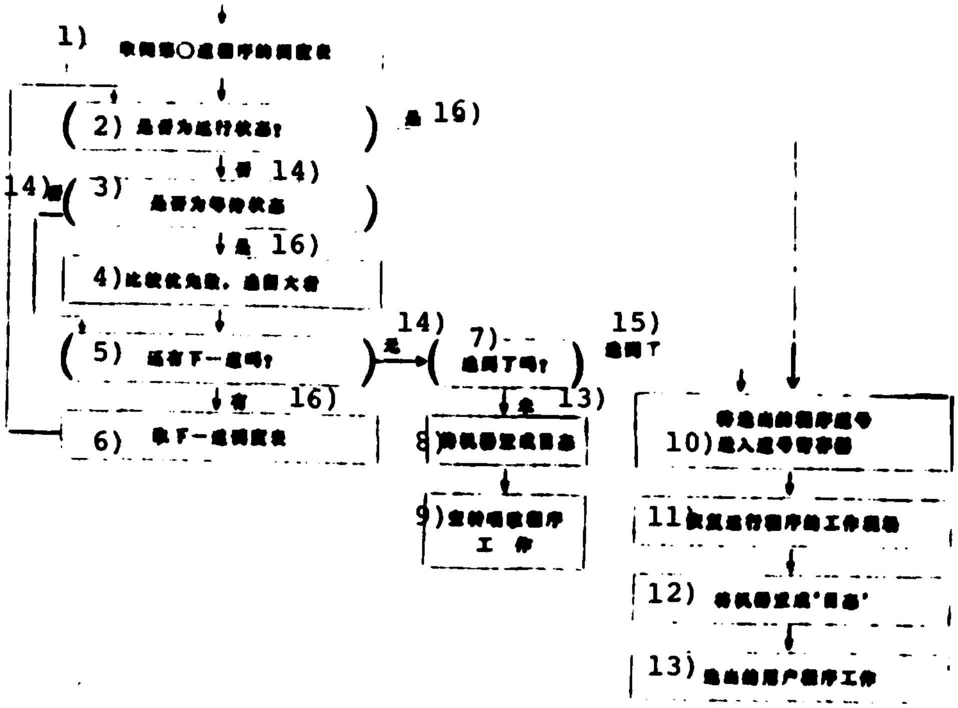
Priority number is a reference for supervision program to select execution program. Each user program is given a priority number which indicates user program execution order. The greater the priority number is, the higher the class is, and the execution will be first.

Priority number can be determined by the user based on the degree of urgency, extent of using external devices, and process capacity. Priority number is given to supervision program when a program enters the machine. During the process of execution, the supervision program can be told to change priority number. Priority number can also be assigned by supervision program based on the characteristics of each user program, such as execution capacity, extent of using external devices, time already spent on execution, waiting time, by using specific formula. Priority number is continuously modified according to the changes of condition inside the machine.

2) Assignment Program

Assignment program is used to determine the execution order of different user programs in the machine. Based on given selection principle, the assignment program can select one channel for execution from among the users in waiting status. So it is also called channel selection program. Assignment program is a part of supervision program. Once "under supervision" it has to go through assignment program to be "out of supervision." The diagram of selecting an execution program by the assignment program is as follows:

In DJS-11, supervision program handles feed-in, suspending and withdrawing of user program. An operator can use operation command through supervision program to control these things.



1) use channel o program assignment schedule
2) Is it in execution status?
3) Is it in waiting status?
4) comparing priority number, select a greater one
5) Is there another channel?
6) get another assignment schedule
7) Has one been selected?
8) turn machine to "print-out"
9) empting singing program, work
10) send channel number of the selected program into
    channel number storage
11) resume execution program working status
12) turn machine to "print-out"
13) selected user program work
14) no 15) selected, 16 yes

BLOCK STRUCTURE LANGUAGE SYSTEM

Chairman Mao once said, "why the problem of mankind is a basic problem and a problem of principle." In order to make DJS-11 widely used by our workers, farmers, soldiers, technicians and scientists to carry out programs of automatization, a block structure language which suits scientific and technological computation has been designed. This language takes care of both function of expression and convenience of writing, it provides more adjustment programs so it is easy to learn and use.

DJS-11 language uses degenerate program as compiling method, object program design is a floating structure and it can accomplish with one scanning. In order to promote the processing speed it must use subscript address degenerate calculation method to deal with subscript variables, which can greatly affect the speed. Combining the characteristics of advance control, optimization of depending on machine should be considered. The compilation structure is simple and the compilation speed and processing speed are great.

1. Block Structure of Source Program

When a problem is described by block structure language, it is made up of one main program segment, several subroutine program segments and function segments. Each program segment consists of:

program segment head
{descriptive statement serial}
{execution statement serial}
{data statement serial}
end

Program segment head gives type and name of the segment. If it is a subroutine program segment or function segment, it also gives type and name of parameter.

End indicates completion of this segment.

Descriptive statement describes quantities used in the program, such as numerical group, function, common numerical group, common block, format, tape, disc. Descriptive statement includes variable address, numerical group, function, tape, disc, common, common numerical, and format statements.

Execution statement includes execution, computation, branch, condition, cycle, assignment, return, code, combination withdrawal statements.

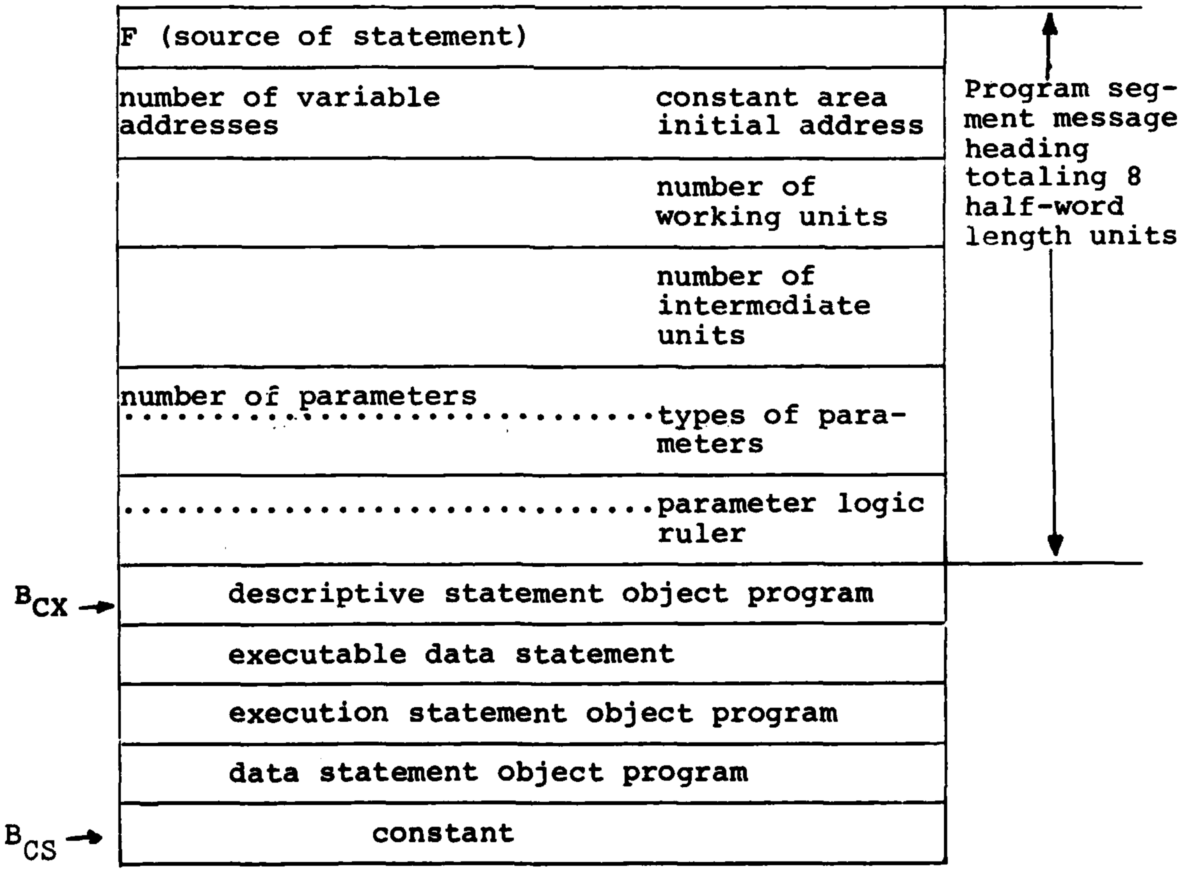
There is no fixed order among the main program segment, subroutine program segment and function segment. But, at the beginning of a whole program, the head of the program must be written and "a completion" at the end. The head shows priority number, request to use external devices and computation status, and "completion" indicates the end of the whole program.
As a computation problem, these three segments are unified unit but each one has its own relative independent and closed characteristics. Basically, the structure of each program segment is the same, but the ways of writing the head are different. For instance, the main program segment has a head, #ZU M; the head of the subroutine program segment is #ZC (name of subroutine program) (parameter) and (type), or #ZC (name of subroutine program).

In order to raise object program execution speed and to widen expression capacity, code expression is introduced into language description.

For convenience in use, employment of degenerate subroutine program segment and function segments is allowed for this language. This language also provides deletable expression, operation variables, description switch, if-switch, tracing, numeration, and tracking facilities for program adjustment. In writing, phonetic Chinese spelling is used after a symbol #, e.g., #ZU means main program, #SZ numeric group. Standard subroutine program, standard function, stored program segment are all headed by \$ which is followed by name, for example, SIN is for sine

2. Floating Structure of Object Program

The object program uses floating type and its common measurement is indicated by absolute address. Each program depends on variable address B_{CX} to float, the local measurement of each program depends on variable address B_{GZ} to float, each intermediate unit depends on variable address B_{ZJ} to float, and each constant depends on variable address B_{CS} to float. Each numerical group uses dynamic distribution. Object program of each program segment has general structure as follows:



	F (source of statement)		Program segment message heading totaling 8 half-word length units
	number of variable addresses	constant area initial address	
		number of working units	
		number of intermediate units	
	number of parameters	types of parameters	
	....	parameter logic ruler	
B_{CX} →	descriptive statement object program		
	executable data statement		
	execution statement object program		
	data statement object program		
B_{CS} →	constant		

The object program is equipped with execution path, when it enters a subroutine or a function segment the execution path moves downward. When a subroutine or a function segment returns the execution path moves up. The structure of an execution path is shown below.

A_l	A_{l-1}	linkage unit
	upper rank program name unit	
	number of main rank variable addresses	
	initial address of main rank executable unit	
	main rank variable address preserved area	
B_{GZ} →	parameter unit	
	function value unit	
	local executable unit	
B_{ZJ} →	intermediate executable unit	
ZJT →	main rank dynamic distribution area	path top

When dynamic numerical group is distributed, path top arrow ZJT moves down. The advantages of using this kind of structure are:

1) The whole program is floating. As long as no common measurement is not included in any one program segment, measurement location is floating. The high quality measurement obtained by compiling with code statement can be fed into the program bank without modification, and it can be linked to other language programs directly. Therefore, by using floating structure, the structure storage bank becomes convenient and it is time saving to compile a program bank.

2) Since whole program depends on four variable addresses to float, object program can be attained with one scanning, advance editing can be omitted. The entire block structure language compiling program is about 8000 half-words long. So, compiling program is brief and compiling time is short.

3) Since the local measurement of each program segment is dynamically distributed downward, a degenerate assignment can be carried out.

4) Due to dynamic distribution of numerical group, internal storage is saved.

However, with this kind of structure to transfer from one program segment to another, those floating addresses must be preserved or revived. Execution speed is then somewhat affected.

3. Degenerate Computation of Subscripted Address

Because of the floating structureof object program, dynamic distribution of numerical group and single scanning of compiler, the computation of subscript address can use simplified dynamic degenerate computation method. It is regulated that the initial value, final value, and step length of cycle are determined by cycle head and cannot be changed in the cycle.

1) The conditions of simplified dynamic degenerative computation are:

a) Subscript expression of subscript variables cannot consist of simple variation, subscript variable, and function. It can only include constants, cycle variables, value parameter and operation variables.

b) Cyclic order of subscript variable is set to be XHN, the highest order value parameter for subscript expression of subscript variable is XJ and the order is XJN.

If XJ appears once in the expression and its coefficient is a constant C, the initial address and increment of subscript variable can be computed outside of XJN cycle. It is said that this subscript variable can be optimized upon XJN cycle.

IF XJ appears twice and its coefficients are not constant, it is believed that the initial address and increment of this subscript variable cannot be computed outside of the XJ cycle. But, if XJH $<$ XHN, the initial address of subscript variable can be computed outside cycle XJN + 1.

If subscript expression does not depend on cycle parameter the address can be computed outermost of the cycle and the increment is naturally zero.

e.g. $\# \; XH \;\; I=1,\; N;$

$\# \; XH \;\; J=1,\; N;$

$A[I,\; I]=B[I,\; J]*C[J]+D[I]+E[2];$

there $D[I],\; E[2]$ can optimize upon I cycle

$A[I,I],\; B[I,J],\; C[J]$ can optimize upon J cycle.

2) Dynamic degenerate computation formula of simplified subscript variable address.

If the numerical group description is

$$\# SZ\; A[J_n, J_{n-1}, \cdots, J_1] \qquad (1)$$

here, $J_i \; (i=1,\cdots,N)$ is an expression.

Subscript variable $A[y_k, y_{k-1}, \cdots, y_1]$ its subscript expression linear depends on innermost order cycle parameter V_t

$$y_k = P_k + C_k V_t \qquad (2)$$

here P_k does not have a parameter larger than V_t expression of simple, subscript variable, and function.

C_k - constant, V_t - inner cycle parameter

$$DIE(A[y_n,\cdots,y_1]) = DIE(A[1,\cdots,1]) + \sum_{K=1}^{n} (y_K - 1) \cdot \Delta_K \qquad (3)$$

Here $\Delta_1 = 1, \; \Delta_K = J_K \cdot \Delta_{K-1}$

Let $V_t = S_t + (N_t - 1) \cdot B_t$

where S_t is a cycle initial value, B_t is cycle step length, N_t is cycle order number.

Thus formula (3) can be rewritten into:

$$DIE(A[y_n,\cdots y_1]) = DIE(A[1,\cdots,1] + \sum_{K=1}^{n} (y_K - 1) \cdot \Delta_K$$

$$= DIE(A[1,\cdots,1]) + \sum_{K=1}^{n} (P_K - 1)\Delta_K + \sum_{K=1}^{n} C_K(S_t + (N_t - 1) \cdot B_t) \cdot \Delta_K$$

$$= DIE(A[y_n,\cdots,y_1]|_{V_t = S_t}) + \sum_{K=1}^{n} C_K(N_t - 1) \cdot B_t \cdot \Delta_K$$

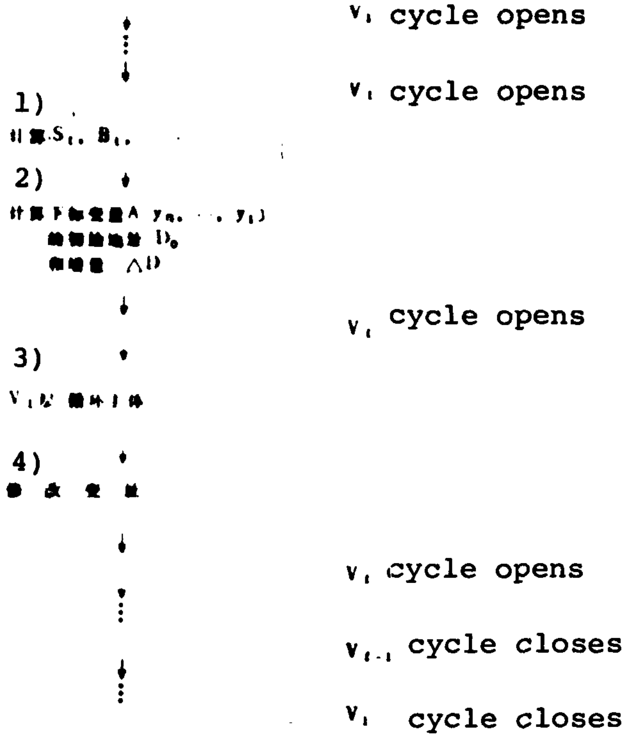
$$= D_0 + (N_t - 1) \cdot \Delta D$$

Here $D_0 = DIE(A[y_n,\cdots,y_1] | V_t = S_t)$

$$\Delta D = B_t \cdot \sum_{K=1}^{n} C_K \cdot \Delta_K$$

No matter how complicated formula S_t, B_t, J_t can be, during execution, their values have already been obtained outside the cycle. If one replaces V_t with S_t and executes "subscript address program" then D_o is attainable. Since V_t is limited to appear once in subscript expression and only one $C_k \neq 0$, thus the compiler program to obtain ΔD is greatly simplified. When ΔD is a constant there is no need to make such a program.

3) The structure of the object program of subscript variable address dynamic degenerate computation.



1) compute

2) compute
subscript variable,
its initial address D_o
and increment ΔD

3) V_t rank cycle main body

4) change address

4. Optimization Upon Machine Instructions

If only subscript address degenerate computation is considered, the object program produced is still quite different from the object program of compilation language. Sometimes their speeds are different by several times because the characteristics of the machine instruction are not fully utilized. When compiling system tries to fully utilize the characteristics of machine instruction, and compresses the number of inner cycle instructions, these instructions get closer to compilation language and the execution speed of object program comes closer to that of compilation language.

It takes 1000 instructions to optimize subscript address cynamic degenerate computation and machine dependent instruction. The optimized solution program is 15 times faster in solving 200th order linear algebraic equation. To solve this equation, it takes compilation language 15.5 seconds and block structure language 22 seconds. If source program of block structure language is well written, and optimization conditions are fully utilized, the execution speed of object program will be faster.

CONCLUSION

The program language of DJS-11 includes block structure language and code compilation language, they result in higher quality, programs so it can be used to compile frequently used programs and other software. Another advantage of code compilation program is the symbolization of command operation and message display it is easy to use.

The DJS-11 machine inspection programs include main machine inspection and external devices inspection. Both are designed for maintenance.

The software of DJS-11 carries 18,000 words, in test, it produces very good results in complicated engineering designs, weather forcasts, data processing of earthquake information, and others. In the treatment of artificial earthquake deviation in petroleum geology surveys the task cannot be accomplished by using computers of small or middle size, but it can have quick results by using the one million cycle large computer.

"In production struggle and scientific experiments, human beings as well as great nature continue to develop and will never stop forever at one level. So, we must increasingly summarize our experiences so as to make some discovery, invention, creation, and progress." At the present time, as a result of the continuous development of computer science, the requirements for software have become more and more and software has become a very important component of the computer. It has especially great potentials when the way of using computers is to be simplified, the function of computers is to be expanded, and the efficiency of using a computer is to be promoted. In the work of machine DJS-11, due to the equipment of multitrack operation, the ability and speed in solving problem by the machine have been indeed promoted; by using language compilation program, the process of solving problems has been significantly shortened, and the importance of software has been primarily demonstrated. Since our using a computer is still at the beginning stage, the function of computer needs to be strengthened and we try to perfect it through experience. As Chairman Mao once "The Chinese people have their will and ability, we must in the near future catch up with or surpass the world level." We work very hard and practice conscientiously so as to make DJS-11 able to contribute greatly to our socialist construction.

DISTRIBUTION LIST

DISTRIBUTION DIRECT TO RECIPIENT

ORGANIZATION	MICROFICHE
A205 DMATC	1
A210 DMAAC	2
B344 DIA/RDS-3C	9
C043 USAMIIA	1
C509 BALLISTIC RES LABS	1
C510 AIR MOBILITY R&D LAB/FIO	1
C513 PICATINNY ARSENAL	1
C535 AVIATION SYS COMD	1
C591 FSTC	5
C619 MIA REDSTONE	1
D008 NISC	1
H300 USAICE (USAREUR)	1
P005 DOE	1
P050 CIA/CRB/ADD/SD	2
NAVORDSTA (50L)	1
NASA/KSI	1
AFIT/LD	1
LLL/Code L-389	1
NSA/1213/TDL	2

ORGANIZATION	MICROFICHE
E053 AF/INAKA	1
E017 AF/RDXTR-W	1
E403 AFSC/INA	1
E404 AEDC	1
E408 AFWL	1
E410 ADTC	1
FTD	
CCN	1
ASD/FTD/NIIS	3
NIA/PHS	1
NIIS	2